Knowledge Representation

Ref: Chapter 10

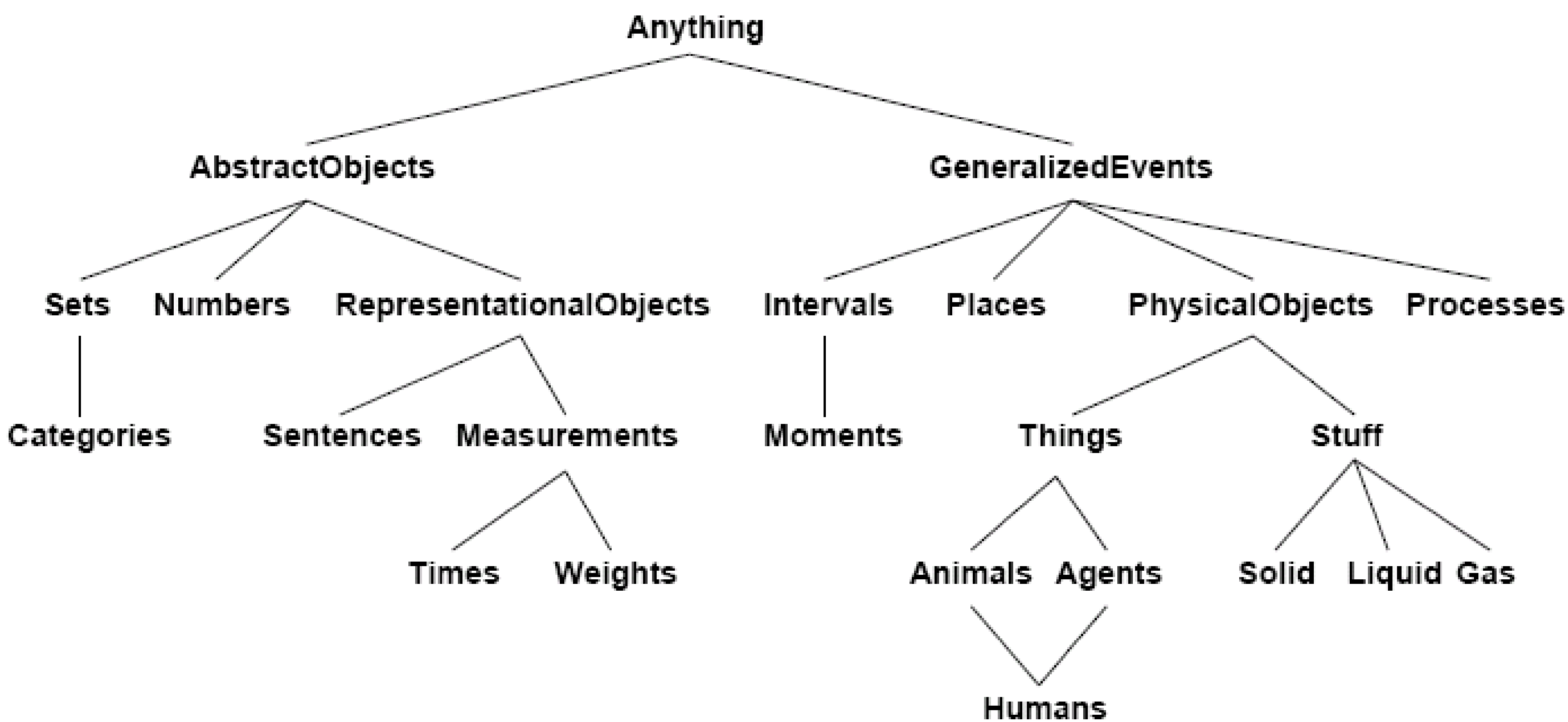
Knowledge Representation

What	**How**
Physical Objects	individuals
Actions	classes
Time	attributes
Beliefs	relations

Ontologies

- Organization of the *world* into hierarchy of categories.
  - a formal, consistent vocabulary.
- Not hard in toy domains
  - Wumpus World: room, wumpus, gold, pit, movement, etc.
- Upper Ontology: broad, general, framework of concepts.
  - Dublin Core, GFO, and others.

An Upper Ontology



Ontological Engineering

- Finding a representation of *a world* so that an agent can:
  - keep track of facts and rules.
  - map sensors to facts and rules.
  - reason (make decisions).
  - reason about reasoning?

Universal Ontology?

- Is there a general ontology on which we can base any special purpose ontology?
  - probably (philosophers have studies this question for centuries).
- The idea is to use a general ontology as a starting point.
  - map domain specific concepts onto the general ontology
  - resolve overlaps/conflicts

Categories and Objects

- KR requires the organization of objects into categories
  - Interaction with the world involves objects.
  - Reasoning involves categories.
- Ex: poker agent uses sensory info to categorize an opponent as "conservative".
  - strategies for playing are governed by the category(s) of opponent.

Categories and Objects

- Categories can be represented in two ways by FOL
  - Predicates: `ConservativePlayer(x)`
  - *Reify* the category as an object:

`Member(x,ConservativePlayer)`

`Subset(ConservativePlayer,BeatablePlayer)`

Organization of Categories

- Categories are related by Inheritance
  - simplifies development and representation of rules and attributes.
  - General rules/attributes apply to broad categories
    - rules/attributes are inherited by subcategories
    - exceptions can easily be expressed as overriding rules/attributes for subcategory.
- Types of relations:
  - subclass/subset , membership, disjoint relations, compositions, etc.

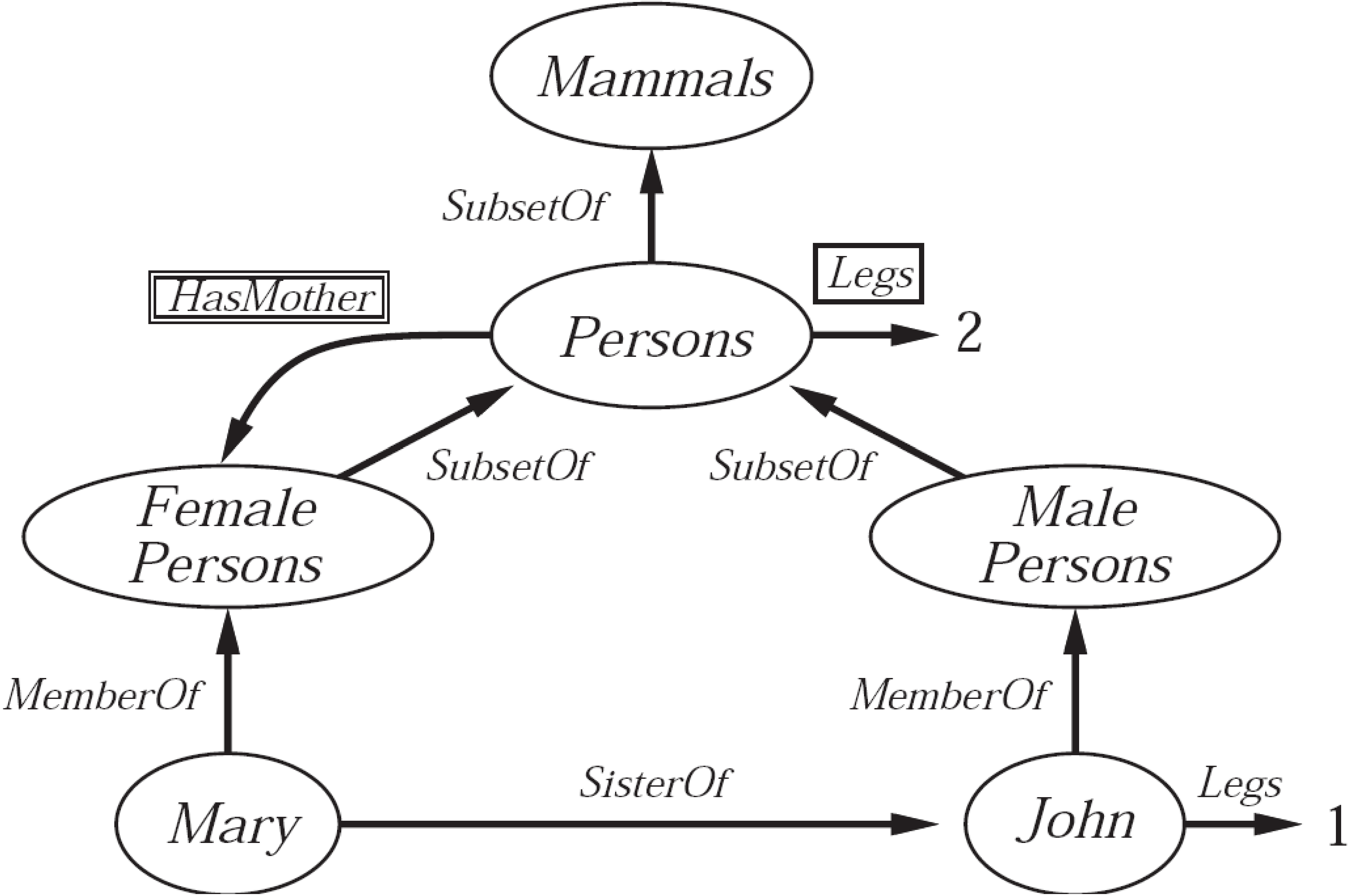
Reasoning System for Categories

- How to organize and reason with categories?
  - Semantic networks
    - Visualize knowledge-base
    - Efficient algorithms for category membership inference
  - Description logics
    - Formal language for constructing and combining category definitions
    - Efficient algorithms to decide subset and superset relationships between categories.

Semantic Networks

- Many variations
  - All represent individual objects, categories of objects and relationships among objects.
- Allows for inheritance reasoning
  - Female persons inherit all properties from person.
- Inference of inverse links
  - SisterOf vs. HasSister

Inheritance Graph/Semantic Network



FOL and Categories

- An object is a member of a category

$$\texttt{MemberOf}(BB_{12},\texttt{Basketballs})$$

- A category is a subclass of another category

$$\texttt{SubsetOf}(\texttt{Basketballs},\texttt{Balls})$$

- All members of a category have some properties

$$\forall x\ (\texttt{MemberOf}(x,\texttt{Basketballs}) \Rightarrow \texttt{Round}(x))$$

Logic and Categories (cont.)

- All members of a category can be recognized by some properties

```
∀x(Orange(x)∧
    Round(x)∧
    Diameter(x)=9.5in ∧
    MemberOf(x,Balls) ⇒
  MemberOf(x,BasketBalls))
```

- A category as a whole has some properties

```
MemberOf(Dogs,DomesticatedSpecies)
```

Category Relations

Two or more categories are ***disjoint*** if they have no members in common:

$$\text{Disjoint}(s) \Leftrightarrow$$

$$(\forall\, c_1,c_2 \;\; (c_1 \in s) \wedge (c_2 \in s) \wedge (c_1 \neq c_2) \Rightarrow$$

$$\text{Intersection}(c_1,c_2) = \{\})$$

`Example: Disjoint({animals, vegetables})`)

Category Relations (cont.)

A set of categories *s* constitutes an ***exhaustive decomposition*** of a category *c* if all members of *c* are covered by categories in *s*:

$$E.D.(s,c) \Leftrightarrow (\forall\, i \;\; i \in c \Rightarrow \exists\, c_2 \;\; c_2 \in s \land i \in c_2)$$

```
ExhaustiveDecomposition({Americans,
 Canadian, Mexicans},NorthAmericans)
```

Category Relations (cont.)

A ***partition*** is a disjoint exhaustive decomposition:

`Partition(s,c)` $\Leftrightarrow$ `Disjoint(s)` $\wedge$ `E.D.(s,c)`

`Partition({Males,Females},Persons)`.

Is `({Americans,Canadian,Mexicans},NorthAmericans)` a partition?

no – some people have dual citizenship

Category Relations (cont.)

Categories can be defined by providing necessary and sufficient conditions for membership

$\forall$ x Bachelor(x) $\Leftrightarrow$ Male(x) $\wedge$ Adult(x) $\wedge$ Unmarried(x)

Physical Composition

One object is a part of another object.

```
PartOf(Bucharest,Romania)
PartOf(Romania,EasternEurope)
PartOf(EasternEurope,Europe)
```

`PartOf` is transitive and reflective:

$$\forall x,y,z\ PartOf(x,y) \land PartOf(y,z) \Rightarrow PartOf(x,z)$$

$$\forall x\ PartOf(x,x)$$

We can infer `PartOf(Bucharest,Europe)`

Physical Composition (cont.)

Often characterized by structural relations among parts.

Example: $Biped(a) \Rightarrow$

$$(\exists l_1, l_2, b)(Leg(l_1) \wedge Leg(l_2) \wedge Body(b) \wedge$$

$$PartOf(l_1, a) \wedge PartOf(l_2, a) \wedge PartOf(b, a) \wedge$$

$$Attached(l_1, b) \wedge Attached(l_2, b) \wedge$$

$$l_1 \neq l_2 \wedge (\forall l_3)(Leg(l_3) \Rightarrow (l_3 = l_1 \vee l_3 = l_2)))$$

Measurements

- Objects have height, mass, cost, ...
  - Values that we assign to these are **measures**
- Combine Unit functions with a number:

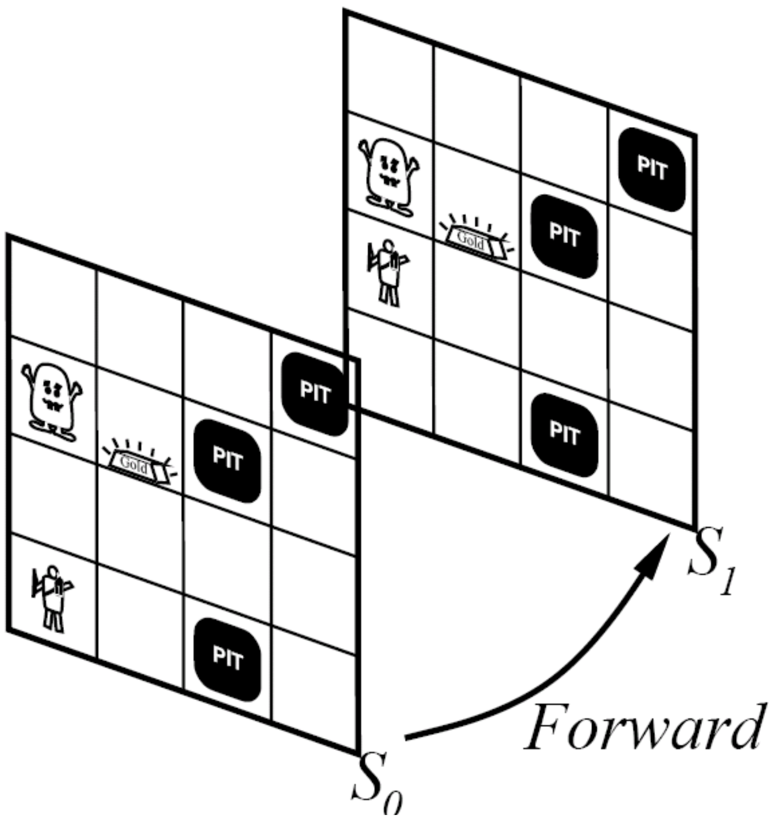
$$\text{Length}(L_1) = \text{Inches}(1.5) = \text{Centimeters}(3.81)$$

- Conversion between units:

$$\forall\ i\ \text{Centimeters}(2.54 \times i) = \text{Inches}(i)$$

- Some measures have no scale: Beauty, Difficulty, etc.
  - Most important aspect of measures: that they are order-able.

Actions and Situations

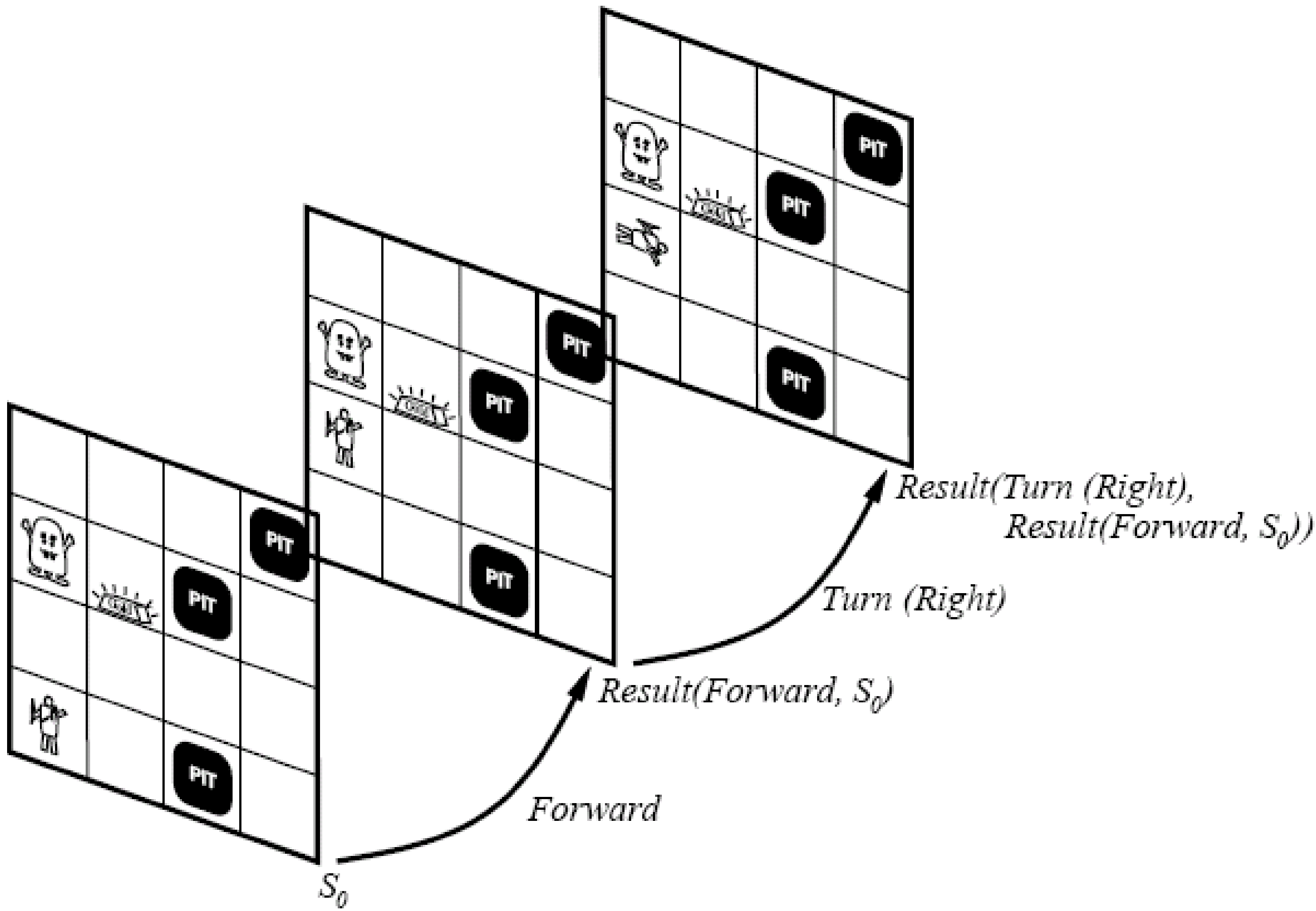


- Reasoning about actions is why we need a KB!
- We need rules that deal with time (for example, the next time step).
- In FOL, it is actually simpler to deal with *situations*
  - states that result from actions

Situation Calculus

- Ontology:
  - situations: logical terms including an initial situation S_0 and all situations that can be generated by applying an action to a situation.
  - *Result(a,s)* is the situation resulting from action *a* in situation *s*
    - *Result(a,s)* is a function.

Wumpus Situations



Fluents

- A *fluent* is a function or predicate that varies from one situation to the next.
  - Wumpus World Examples:
    - Location
    - $\neg Holding(G_1, S_0)$
      - *gold G_1 is not held in the initial situation*

Atemporal/ Eternal

- Predicates/functions that always hold (regardless of the situation).
- Many facts are atemporal.
- Wumpus Examples:

  $Gold(G_1)$

  $LeftLegOf(Wumpus)$

Situation Calculus Tasks

- *Projection task:* an Situation Calculus agent should be able to deduce the outcome of a sequence of actions.

- *Planning task:* find a sequence that achieves a desirable effect.

Describing Change

- Situation calculus requires two axioms to describe change:
  - Possibility axiom: when is it possible to do the action

$$At(Agent,x,s) \wedge Adjacent(x,y) \Rightarrow Poss(Go(x,y),s)$$

  - Effect axiom: describe changes due to action

$$Poss(Go(x,y),s) \Rightarrow At(Agent,y,Result(Go(x,y),s))$$

Describing what doesn't change

- This is called the *frame problem*
- Typically most things don't change when an action takes place.
  - need a way to explicitly represent which things don't change.
- Frame Axioms:
  - describe non-changes due to actions
  - Example: what moves with the wumpus:

$$At(o,x,s) \wedge (o \neq Agent) \wedge \neg Holding(o,s) \Rightarrow At(o,x,Result(Go(y,z),s))$$

Representation Frame Problem

- If there are F fluents and A actions then we need AF frame axioms to describe objects that are stationary unless they are held.
  - We write down the effect of each actions
- Solution; describe how each fluent changes over time
  - Successor-state axiom:

$$Pos(a,s) \Rightarrow$$
$$(At(Agent,y,Result(a,s)) \Leftrightarrow$$
$$(a = Go(x,y)) \vee (At(Agent,y,s) \wedge a \neq Go(y,z))$$

  - Note that next state is completely specified by current state.
  - Each action effect is mentioned only once.

Other Problems

- How to deal with secondary (implicit) effects?
  - If the agent is carrying the gold and the agent moves then the gold moves too.
  - Ramification problem
- How to decide EFFICIENTLY whether fluents hold in the future?
  - Inferential frame problem.
- Extensions:
  - Event calculus (when actions have a duration)
  - Process categories

Mental Events and Objects

- So far, KB agents can have beliefs and deduce new beliefs
- What about knowledge about beliefs? What about knowledge about the inference process?
  - Requires a model of the mental objects in someone's head and the processes that manipulate these objects.
- Relationships between agents and mental objects: believes, knows, wants, …
  - Believes(Lois,Flies(Superman)) with Flies(Superman) being a function … a candidate for a mental object (reification).
  - Agent can now reason about the beliefs of agents.

Internet Shopping Agent

- A Knowledge Engineering example
- An agent that helps a buyer to find product offers on the WWW.
  - IN = product description (precise or not)
  - OUT = list of webpages that offer the product for sale.
- Environment = WWW
- Percepts = web pages (character strings)
  - Extracting useful information required.

Shopping Agent

- Find relevant product offers

  RelevantOffer(page,url,query) ⇔
  Relevant(page, url, query) ∧ *Offer(page)*

  - Write axioms to define Offer(x)
  - Find relevant pages: Relevant(x,y,z) ?
    - Start from an initial set of stores.
    - What is a relevant category?
    - What are relevant connected pages?

- Compare offers (information extraction)
- See the text for more details...